TASK FORCE WEB

User Facing Services (UFS) Interfaces

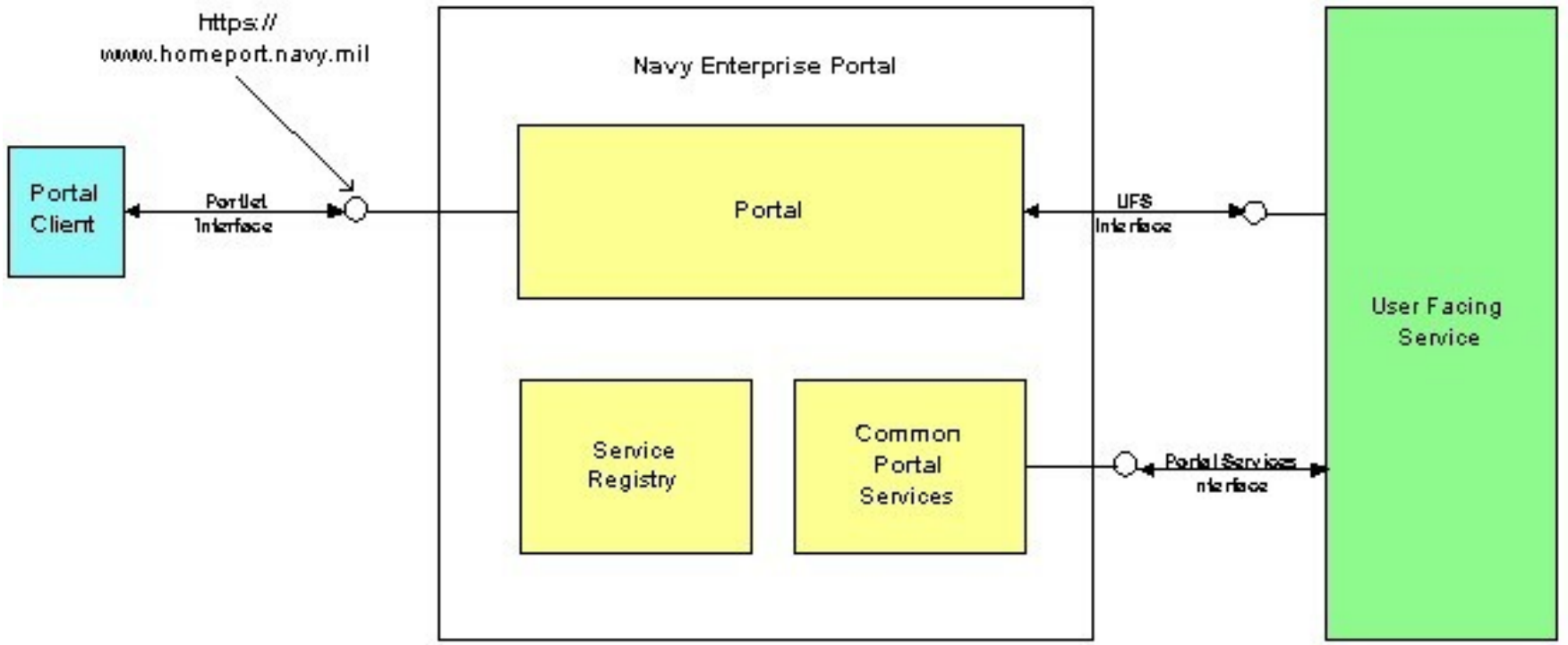
9 July, 2002

Michael Donovan, EDS
NMCI/ISF Enterprise Technology

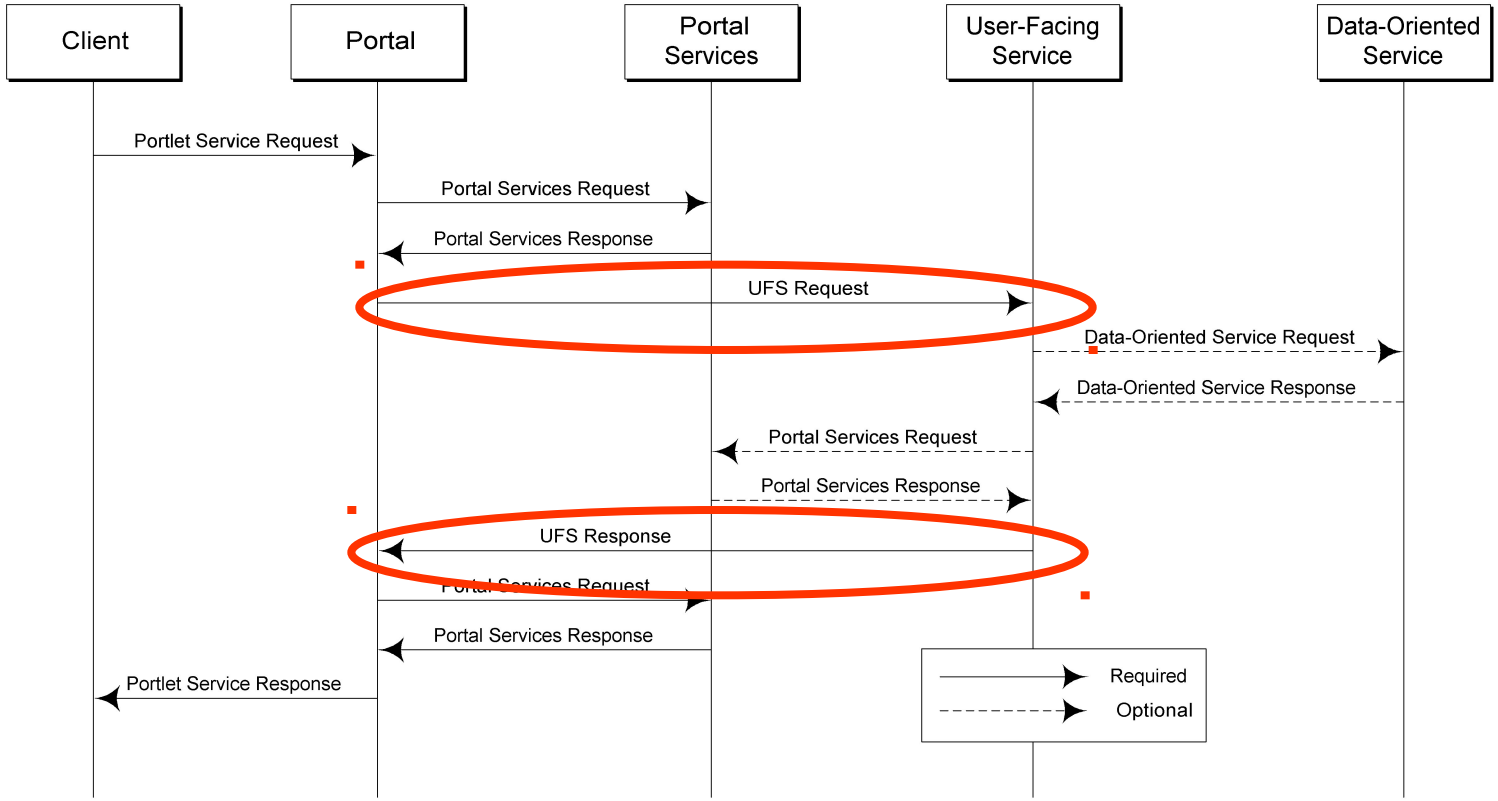
Agenda

- Interface Overview
- UFS Invocation (Portlet Interface)
- UFS Response (UFS Interface)
- Optional Portlet Interface Features
- Optional UFS Interface Features
- Interface Documentation

NEP Architecture



Interface Execution



UFS/Portal Interaction

- The *Portlet Interface* is used to invoke the UFS each time a user brings up a workspace containing the portlet.
  - An HTTP request is sent to the UFS based on the information supplied in the Service Registry. The invocation method is either a normal HTTP request or a SOAP envelope over an HTTP binding, based on the tModel associated with the UFS.
- The *UFS Interface* is the response to the portal request.
  - An HTTP response is returned to the portal containing the HTML or XML/XSL used to render the content for the user. For Web Services (WSDL) based responses, the content is contained in a SOAP envelope.

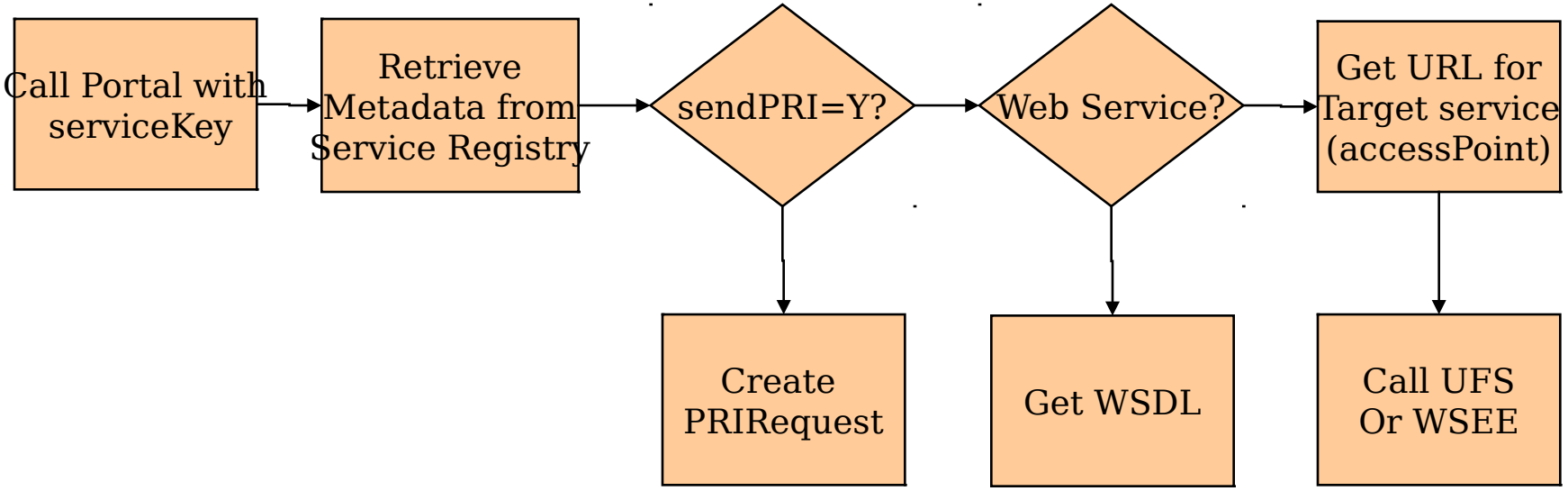
Portlet Interface

- When a UFS is published in the NEP, it is coded as a call to the PortalConnector servlet. Invocation of the UFS looks like the following, taking all defaults:

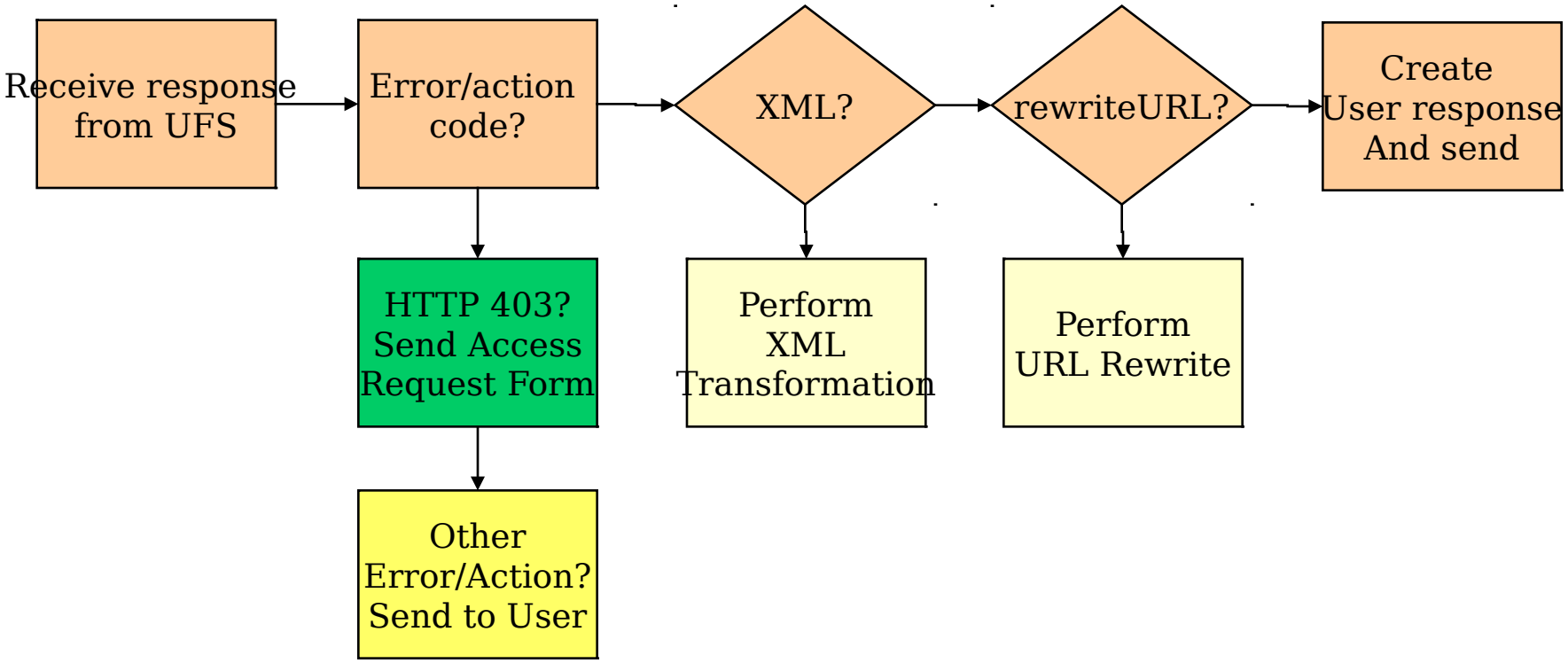
  https://www.homeport.navy.mil/servlet/PortalConnector?serviceKey=0A837A83-D8DE-43F4-A281-EC2981CB03A&templateKey=0

  - The serviceKey is the service registry bindingKey for the service, from the UDDI specification.
  - The templateKey is provided by the portal engine and identifies the looks and feel template the user has selected.
- The developer may choose to accept the default behavior, or provide the portal administrator with optional parameters.

Portlet Interface
Default Behavior



Portlet Interface Default Behavior



UFS Interface

- The UFS Interface is a standard HTTP 1.1 compliant request/response pair as defined in *RFC 2616, Hypertext Transfer Protocol – HTTP/1.1.* Only the HTTP binding is currently supported.
- There are two types of messages:
  - Messages with a SOAP envelope (wsdlSpec tModel)
  - Messages without a SOAP envelope (http transport tModel)

UFS Interface Default Behavior

- For normal interaction, the requirement is an HTTP response containing the HTML, XML or XML/XSL to generate the content presentation. In other words, a normal web response.
- The response must contain a SOAP envelope for WSDL interfaces

Portlet Interface Optional Features

Parameter	Value	Description
insertStyle	[Y | N]	Inserts the appropriate Portal cascading style sheet (CSS) into the HTML UFS output before sending to the Portal Client.
renderXML	[Y | N]	Controls whether the Portal will attempt to render XML using an XSLT stylesheet reference imbedded in the XML document. Setting this to N will allow a service to pass the raw XML to the client to support client side rendering.
rewriteURL	[Y | N]	Controls whether the Portal will attempt to rewrite URL references in the return stream to proxy all requests back through the portal.
generateCookie	[Y | N]	Controls whether the Portal will generate a session cookie with the information necessary to allow re-invocation. This is intended to be used when a forward proxy is used instead of the URL rewrite

Portlet Interface Optional Features

Parameter	Value	Description
bindingType	[HTTP | WSDL]	The default behavior of the Portal is to use a WSDL binding (tModel) if one exists, otherwise the accessPoint is called using an internal HTTPClient. Specification of this parameter can be used to override this behavior.
contentMimeType	[mime-type]	Overrides the default behavior of the Portlet Interface process flow. In general, if the portal can identify XML data, then rendering is performed. If HTML, then URL Rewrite is performed. If the response from the service does not contain a mime type and is consistent, then specifying the mime-type on this parameter can increase efficiency as the logic that attempts to determine the mime-type can be avoided.

Portlet Interface
Optional Features

Parameter	Value	Description
sendIdentity	[Y | N]	Controls whether the Portal will copy the HTTP authentication header from the original client request to the request generated by the portal and passed to the service being called.
userName	[identity]	Deprecated. The value supplied will be ignored. The identity of the user will be derived from the HTTP authorization header
sendPRI	[Y | N]	Indicates whether or not the PRIdata message is added to the request sent to the service. The method used to pass the PRIdata is dependent upon the type of invocation used. The PRIdata message is attached to the HTTP header for standard HTTP calls and if SOAP is used, is imbedded in a SOAP header (ACTOR=NONE).

Portlet Interface Optional Features

Parameter	Value	Description
uddiURL	[URL]	The default behavior of the WSEE service invoked by the Portal is to lookup all bindingKey references in the NEP Enterprise Registry. If the service is defined in a public UDDI registry, the URI to the inquiry interface must be specified here.
Operation	[operation-name]	The operation name in the called web services definition language (WSDL) file to be executed.
operationMessageParts	[keyword:value|…]	The messages passed to the operation to be executed.
bindingPort	[port-name]	The name of the instance of the service as defined in the WSDL file. If multiple endpoints are defined for access to the service and a specific instance must be used, this parameter allows override of the default behavior.

UFS Interface
Optional Features

- Retrieve and use Common Identity (sendIdentity=Y)
- Authenticate and authorize user to access service resources
- Retrieve, parse and use metadata from the PRI (sendPRI=Y)
- Integrate portal look and feel CSS styles
- Perform application domain specific functions
- Interact with other services (DOS, RDBMS, etc)
- Perform XML rendering (renderXML=N)
- Perform URL rewrite or proxy operations (rewriteURL=N)
- Generate and return error codes (HTTP, PRI Response or SOAP Fault)

Portlet Interface Examples

- Request invocation as a web service
  https://www.homeport.navy.mil/servlet/PortalConnector?serviceKey=0A837A83-D8DE-43F4-A281-EC2981CB03A&templateKey=0**&operation=getQuote**
  - This invocation will call the service from the private registry and generate a form to request any required messages (parameters)
- Get PRIData and Common Identity
  https://www.homeport.navy.mil/servlet/PortalConnector?serviceKey=0A837A83-D8DE-43F4-A281-EC2981CB03A&templateKey=0**&sendIdentity=Y&sendPRI=Y**
  - This invocation requests that the PRIData XML data be sent in either HTTP header or SOAP header format and that the HTTP authorization header is attached.

Interface Documentation

- Contract Programming
  - Concept developed by Bertrand Meyer for Eiffel OO language
  - Contracts are written between two parties when one of them (the *supplier*) performs some task for the other (the *client*). Each party expects some benefits from the contract, and accepts some obligations in return. Usually, what one of the parties sees as an obligation is a benefit for the other. The aim of the contract document is to spell out these benefits and obligations.

NEP Interface Contracts

Preconditions	Defines all of the requirements that must be satisfied before an interface is used. These preconditions may include both real-time software execution conditions and policy conditions
Service Request	Describes the request made of the service. Includes format, parameters, values and if required or optional
Service Responsibilities	Describes what the service is expected to accomplish. May vary based on content of the Service Request
Service Response	Describes the response returned by the service. Multiple types and formats may be described based on actions taken by the service.
Postconditions	Defines the assumptions that can be true after processing by the service has completed. Includes any error codes defined for the service.